

OOP

Caroline Lemieux

March 7th 2019

Announcements

Homeworks + Labs

Homework 5 is due Friday 3/8

Lab 6 is due Friday 3/8

Projects

Ants due next Thursday! (Midpoint due next Monday)

Midterm 2 Approaching 🗓️

Date TBD, March 19th or 20th. Start studying!

Get help with concepts @ OH/Lab/CSM/1-1 Advising

Concept Check

1. What is the relationship between a `class` and an `instance`?

```
class Professor:
    degree = "PhD"

    def __init__(self, name):
        self.name = name

    def lecture(self, topic):
        print("Today we're learning about " + topic)

dumbledore = Professor("Professor Dumbledore")
```

Concept Check

1. What is the relationship between a **class** and an **instance**?

```
class Professor: class  
    degree = "PhD"
```

```
    def __init__(self, name):  
        self.name = name
```

```
    def lecture(self, topic):  
        print("Today we're learning about " + topic)
```

```
instance dumbledore = Professor("Professor Dumbledore")
```

Concept Check

1. What is the relationship between a **class** and an **instance**?

```
class Professor:  
    degree = "PhD"
```

```
    def __init__(self, name):  
        self.name = name
```

```
    def lecture(self, topic):  
        print("Today we're learning about " + topic)
```

```
instance dumbledore = Professor("Professor Dumbledore")
```

creating an **instance**

Concept Check

2. What is the difference between an [instance attribute](#) and a [class attribute](#)?

```
class Professor:
    degree = "PhD"

    def __init__(self, name):
        self.name = name

    def lecture(self, topic):
        print("Today we're learning about " + topic)

dumbledore = Professor("Professor Dumbledore")
```

Concept Check

2. What is the difference between an **instance attribute** and a **class attribute**?

```
class Professor:
    degree = "PhD"

    def __init__(self, name):
        self.name = name

    def lecture(self, topic):
        print("Today we're learning about " + topic)

dumbledore = Professor("Professor Dumbledore")
```

class attribute: all Professors have PhD's

instance attribute: each Professor has their own name

Concept Check

3. What is the difference between a [class method](#) and a [function](#)?

```
class Professor:
    degree = "PhD"

    def __init__(self, name):
        self.name = name

    def lecture(self, topic):
        print("Today we're learning about " + topic)

dumbledore = Professor("Professor Dumbledore")
```


Concept Check

3. What is the difference between a **class method** and a **function**?

```
class Professor:
```

```
    degree = "PhD"
```

```
    def __init__(self, name):
```

```
        self.name = name
```

```
    def lecture(self, topic):
```

```
        print("Today we're learning about " + topic)
```

```
dumbledore = Professor("Professor Dumbledore")
```

method: only Professors can lecture

Concept Check

3. What is the difference between a **class method** and a **function**?

```
class Professor:  
    degree = "PhD"
```

```
def __init__(self, name):  
    self.name = name
```

method: only Professors can lecture

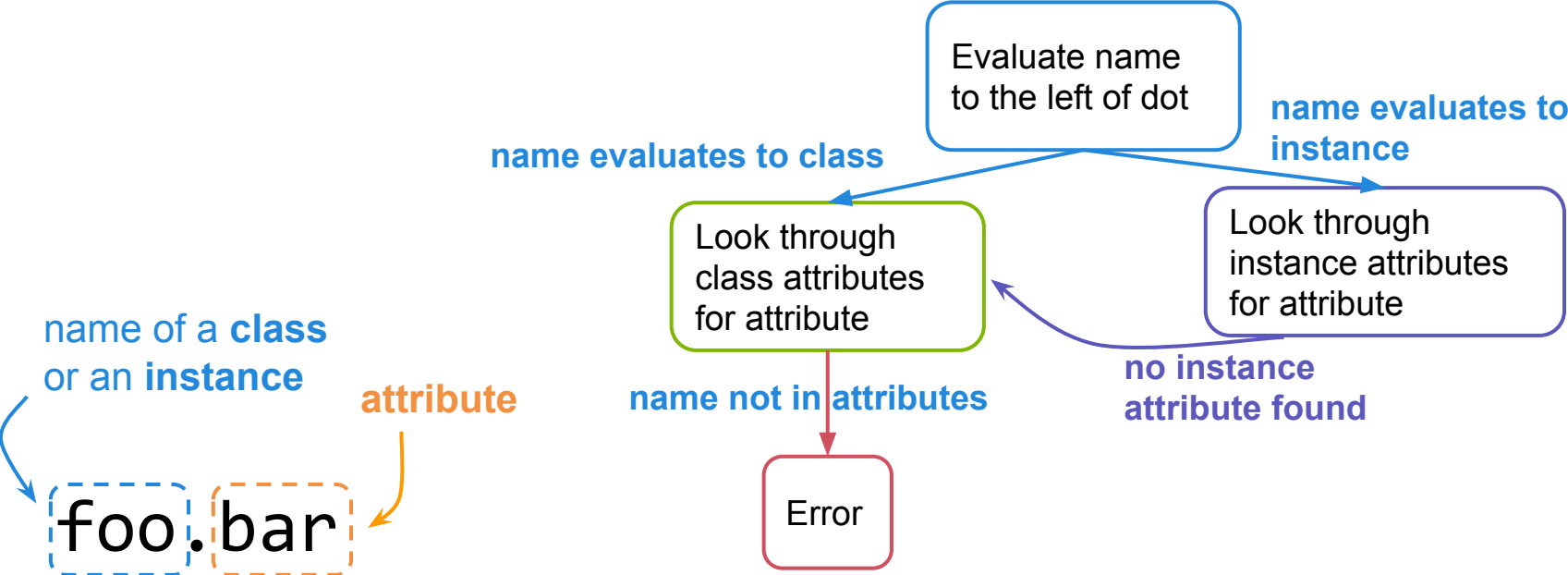
```
def lecture(self, topic):  
    print("Today we're learning about " + topic)
```

```
dumbledore = Professor("Professor Dumbledore")  
dumbledore.lecture("general wisdom") bound method  
Professor.lecture(dumbledore, "general wisdom") function
```

Concept Check - Summary

1. What is the relationship between a **class** and an **instance**?
 - **An instance is a single object belonging to some class.**
2. What is the difference between an **instance attribute** and a **class attribute**?
 - **An instance attribute is specific to a certain instance.** They can only be accessed through that instance. Changing an instance attribute of one instance does not affect the instance attributes of other instances.
 - **Class attributes are shared among all instances of a class.** They can be accessed either using the class name or through an instance. Class attributes can only be changed by accessing it using the class name.
3. What is the difference between a **class method** and a **function**?
 - **A class method is a function that belongs to a class and can only be called on an instance of that class.** A function has no association with any objects.

Dot Notation Lookup



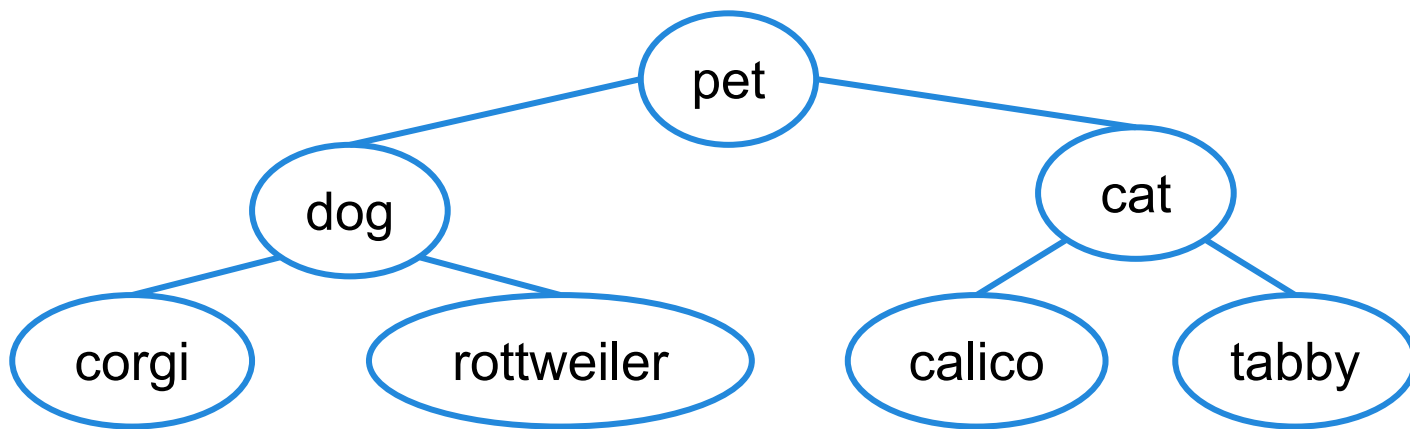
Attendance

links.cs61a.org/caro-disc



Inheritance

- sometimes you have different classes that have very similar attributes and methods!



- **inheritance** allows objects to have all the attributes of another class

Inheritance

```
class Dog(Pet):
```

class signature: class(superclass)

```
    def __init__(self, name, owner):  
        Pet.__init__(self, name, owner)  
        self.tricks = []
```

overriding a method:
redefining a method
that was inherited

dogs are just like any other
pet, but they also have a set
of tricks!

```
    def talk(self):  
        print(self.name + ' says woof!')
```

dogs can learn
tricks! (cats can't...
boooo)

```
    def learn_trick(self, trick):  
        self.tricks.append(trick)
```